Cache Memory Organization: A Performance and Acquisition Variable?

Idris Boundaoni

In fulfillment

of

Department of Computer Science
University of Maryland GC
ITEC 625 9040 Computer System Architecture
Professor Sandra Cirel DeLaus

July 30, 2024

Abstract

Cache memory technologies are essential components in modern computer systems. The use of the cache memory architecture has considerably improved the average memory access time and performance of the modern computer system. Cache is a chip-based, short-term memory where most-accessed data, operand and instructions are stored. This study seeks to provide computer acquisition information on cache memory via investigating the nature, design, and positive impacts of cache memory organization on computer system performance. Besides, it shows the merits and demerits of each cache memory architecture by taking into account disparate scenarios. Cache memory is not only the main consideration when acquiring a computer system. Hence, the use of cache memory does provide some definitive ideas relative to execution speed and performance of the computer system. However, the acquisition of a specific computer system must be contingent upon individual and business core requirements and needs.

Introduction

Over the past decades, computers have become an indispensable and ubiquitous part of the basic human-life activities, economic and political operations. This has called for the need to discover novel architecture to eliminate von Neumann bottlenecks with the view to enhancing throughput between the processing units and memory (Parrilla-Gutierrez et al.,2020). Contemporary computers have come to rely centrally on random-access memory (RAM) as the main memory with two core parts, static RAM (SRAM) and Dynamic RAM (DRAM). DRAM, which is deployed in personal computers, is comparatively cheaper and highly integrated due to its lower energy (power) consumption. DRAM refreshes periodically to eschew data loss. SRAM is specifically designed to run on servers or special or supercomputers. SRAM is very fast however it has a less integration rate (Wang and Ledly,2014). The operational relationship between the

memory units and processor is one of the core parameters used to determine the high-performance status of the computer. Hence, a myriad of memory architectures was tried and tested until the 1968 introduction of the cache memory by IBM. Since then, the evolution of cache memory has seen rapid improvement and enhancement in the computer's processing capability. This has led to good performance and reduction in the energy consumption needs of modern computers (Díaz Álvarez et al.,2015).

This study is designed and conducted to achieve multiple objectives. The primary objective is to comprehensively investigate cache memory organization and operation by emphasizing its contribution to the performance in modern computer systems. Thus, it seeks to examine how the introduction of multiples caches, L1, L2, L3 and L4, has really reduced access time and improved system performance. Besides, the study will make purchase recommendations for pristine sporting good enterprise, our client for the past decades. Furthermore, the study findings can be utilized as a foundation for future study and development in the arena of cache organization and operation systems.

Feature of computer memory organization

The organization of the memory is a key computer design issue. The design has some constraints which can be summed up in terms of capacity, access time, and cost. These three constraints can be expressed in how-much, how-fast, and how-expensive questions. The how-much question relates to the capacity of the memory (Stallings,2013). The how-fast question addresses how fast the memory must be able to keep up with the processor. The cost question is the cost of the

memory, which must be reasonable in relation to other components of the computer or the overall price of the computer (Yadin,2020).

Furthermore, computer memory is made up of memory systems based on location, capacity and unit of transfer (Stallings, 2013 and Yadin,2020). First, the location defines either external or internal nature of the memory. Internal memory refers to the main, registers, or cache memories, the target of this study. External memory is made up of all peripheral storage devices, including disks and tapes. Second, capacity refers to how much data volume can memory space holds. Internal memory capacity is typically expressed in terms of byte, where 1 byte=8bits), or word, where the commo word lengths are expressed in terms of 8, 16, or 32 bits. External memory, on the other hand, is typically expressed in terms of bytes. Third, unit of transfer refers to the mode of transfer, which determines the units of electrical line into and out of the memory units (Stallings, 2013 and Yadin,2020).

Further, over the past decades, a variety of memory technologies faced the dilemma of joggling faster access time, memory capacity, and cost. Computer designers were confronted with three computer design juggernauts (Yadin,2020):

- The memory system that supports faster access time but at a greater cost per bit.
- The memory system that supports large-memory capacity with a smaller cost per bit.
- The memory system that supports greater-memory capacity with a slower access time.

To resolve this bottleneck and design dilemma, computers designers ushered in memory technology that supports large-capacity memory at comparatively low cost per bit (Stallings, 2013). This led to the design of computer memory hierarchy, where the position of each memory

module level determines its access times, cost, and capacity, and thereby determines the system performance.

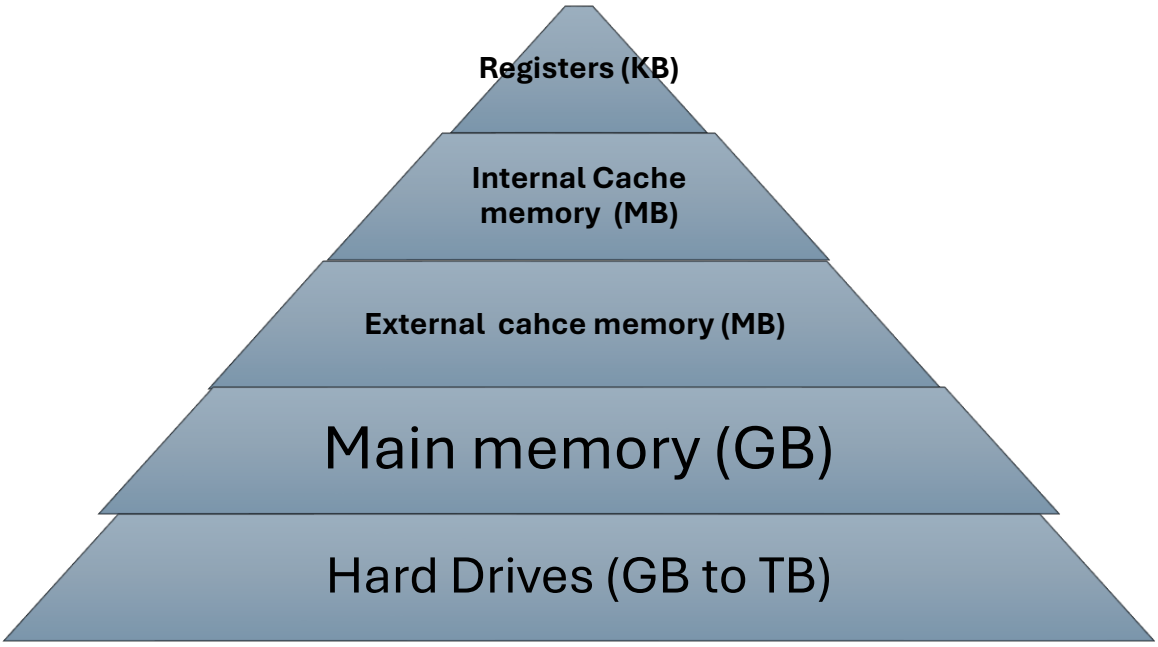


Figure: 1.0 Memory hierarchy

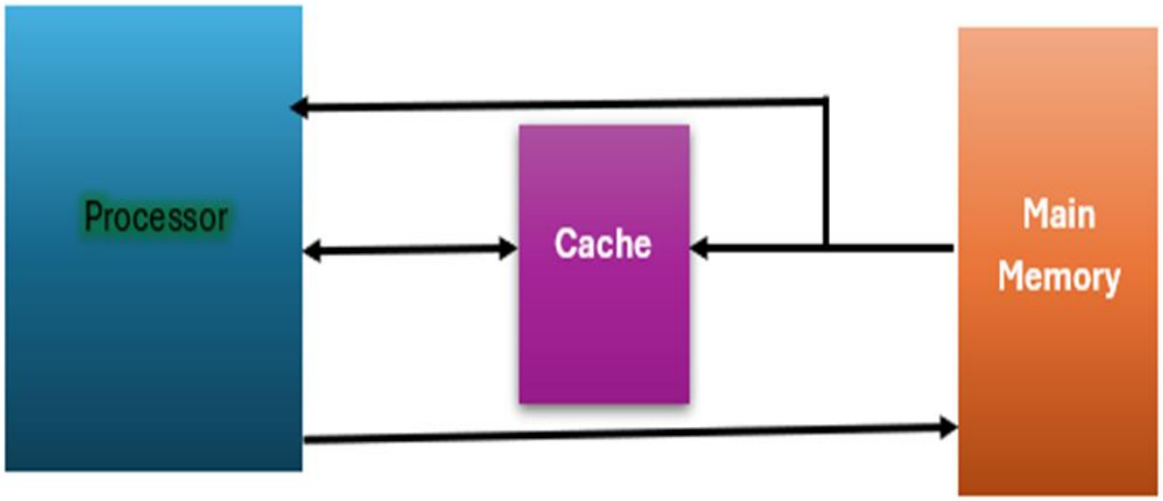
The memory hierarchy illustrates the following information as one goes down the ladder of the memory pyramid: (1) decrease in the cost per bit; (2) increase in the memory capacity of the system;(3) increase in the access time rate; (4) decrease in the frequency of the processor's capacity to access the memory (Stallings,2013). In so doing, the memory hierarchy designers trade off larger, cheaper, slower memories for smaller, more costly, faster memories. Thus, there is a minimal interaction between the processor and the main memory due to the introduction of cache memory into the computer system architecture. The processor reads from the memory only when the data requested from the cache is not available.

Performance of computer memory is dependent on three parameters (Stallings,2013): access time (latency), memory cycle time, and transfer rate. Access time refers to the time it takes for the system to perform read or write operations. Memory cycle time is the access time plus any extra time needed before a second or different operation access can initiate (Mano,1993 and Yadin,2020). The transfer rate refers to the rate at which data can be transferred into or out of a memory module.

Cache Memory

Cache is a temporary memory where most accessed programs and files reside. Thus, it is a comparatively small memory for hoarding instructions and data currently within the processor or that will be required for future execution (Yadin,2019). The cache memory is faster than random access memory of the computer hence the computer uses it when it needs to access data, operand, or instruction rapidly. The cache, sitting between the main memory and the central processing unit (CPU), is the first go-to when the processor needs data or instructions. If the data is found, then the process is very fast. If not, then the processor will literally have to make a trip

to the store of the main memory to bring the data for processing and execution. This situation is



illustrated below in figure 1.1 above:

In other words, from figure 1.1 above, the CPU will always search for a specific data operand, data, or instructions in the cache memory first. If it is there, then access time is decreased which in turn increases process time and performance. If it does not find it there, then it will read it from the main memory. This increases process time and reduces performance of the system. The mapping process, a situation in which data is transferred from the main memory to the cache, is always designed as part of the entire cache organization to decrease access time and improve system performance (Yadin,2020; Mano, 1993; Stalling 2013).

Evolution of Cache Memory Architecture

Cache memory was not originally part of the computer system. The von Neumann architecture was cache memory absent until IBM introduced it in 1968 to boost processing performance and powers of its mainframe computer 360/85. The IBM 360/85 had only L1 cache with a size range of 16-32 kB. IBM minicomputer did not come with cache memory until 1978, a decade after the introduction of cache memory in IBM 360/85. The IBM minicomputer cache size was 1kB. In

1989, Intel introduced an L**1** cache memory in its Intel 80486 personal computers with a cache size of 8 kB (Stallings, 2013).

A novel breakthrough in cache memory architecture transpired in 1993 when Intel introduced L2 cache into the modern computer realm. L2 first appeared in the early series of single-core CPUs for desktops, laptops and entry-level servers by Intel. Hence, other manufacturers designed and distributed computers with two cache memory architecture, L**1** cache and L**2** cache, enhance rapid processor capabilities. In 1999, Cache memory organization took a brilliant turn when another cache level was ushered in into the computer architecture and distribution. This time L3 was procreated into the computer architecture and industry. It first appeared in PowerPC 620 with remarkable and effective cache size of 2 MB (Stallings,2013).

Evaluation of Cache Memory's Effectiveness

Even though there are currently three standard types of cache memories running in modern computers, the measurement of their effectiveness is based on the hit radio (Mano,1993). The hit ratio, the yardstick for cache memory performance, is calculated from the hit rate and the miss rate. The hit is a satiation where the data or instructions needed are indeed residing in the cache. The hit rate is therefore the percentage of cases found data or instructions in the cache. In other words, the hit rate is the likelihood of finding the requested operand or instructions in the cache memory (Yadin,2019 and Mano,1993).

A miss is a situation in which the data needed by the processor is not found in the cache. Hence, the miss rate is the probability of not finding the data being requested by the processor in the cache (Yadin,2020). A high hit ratio is indicative of the CPU access of the cache instead of the main memory. This showcases the effectiveness of the cache memory.

On the other hand, a low hit ratio means that the CPU goes to the main memory most of the time to access requested data not found in the cache memory. Hence, the instruction execution will be slowed down. In other words, if the data needed by the processor is not in the cache memory, then the processor needs to wait. A missing operand means that the instruction is delayed until the operand is transferred from the main memory. However, if the missing data is an instruction, then the processor needs to stop. It will then re-issue the instruction whenever it is issued from the main memory. This is what is called missing penalty, the extra waiting time of the processor. In other words, the missing penalty is the amount of time that is wasted when data requested is not available in the cache memory (Mano,1993 and Yadin, 2020).

To mitigate the cases of missing penalties, the modern computer system is engineered with several buckets of cache memory, L1, L2 and L3 caches and even L4. All these buckets of cache can reside in a single computer and in this case the search for specific data or instruction is hierarchical or pyramidic. L1 cache is designed to lower access time, and it is integrated into the CPU. It is relatively small and thereby resides very closely to the processor. L1 cache memory may be in two parts where one part holds instructions and the other holds data. L2 cache memory purpose is to enhance the hit rate to improve the hit ratio (Yadin, 2020 and Knerl, 2021). L3 cache memory relatively recent. It has become ubiquitous in the era multi-core processor implementations. L3 cache memory is aimed at average access time. It is relatively large cache, for instance Itanium third generation processor supports a large L3 cache (Rusu and Cherkaur,2004). ISA-16 core processor has 3 cache models. The three levels of cache may use still serial access approach where L1 cache is search ed first, and if the data is not found, L2 will be searched, and so on. A high-capacity L4 currently exists in the computer architecture and manufacturing ecology. Last-level

cache architecture (LLC), also known as L4, has been introduced as DRAM and Spin Transfer torque (STT-DRAM technologies. DRAM LLC has higher energy consumption issues while STT-RAM has a lower write endurance (Hameed et al.,2019).

Future Trends in Cache memory Organization

Caches were originally ushered in as single cache. However, more recently, the use of multi-cache has become the norm due to relentless pursuit of performance (Rusu and Cherkaur, 2004). This has really altered how cache memory is designed and introduced either in united caches or split caches. It is possible to due to improved logic density to create on-chip cache which reside exactly where the processor resides with the viewing to reducing processor's bus activities, speed up access and execution time, and increase system performance (Yadin,2020; Fide and Jenks,2008). The use of on-chip cache did not lead to the demise of the off-chip cache- it led to the design of two-level cache where L1 is designated as external cache and L2 designated as external cache, which enabled rapid data access in a zero-wait state via fastest bus transfer.

The merits of contemporary cache design in the areas of L3 and L4 are noteworthy. For instance, IBM Telum Processor uses new horizontal cache persistence algorithms to be programmed L2 caches to serve as system wide L3 and L4 caches (Berger et al., 2023). ISA-16 core processor has 3 caches models with sharing property of L2 and L3 cache banks. This is designed to reduce the area occupied by the caches, locate large caches very close to the processor accessing it, and reduce cache capacity misses (Sibai,2008). The J90 supercomputer developed during the 1990s had several processors (8,16, or 32) with each having its own cache memory (Yadin,2020). The T3E supercomputer too has many processors and each processor being served by a cache memory.

As the technology develops, efficient technology of the cache memory will emerge. Memory price will continue to subside, and this positively impacts the cache architecture and deployment. Hence, there could be a single cache serving each processor in the PC market.

Conclusion

The acquisition of the new computer and servers for the company should not be based on only the information and facts on the performance presented in this study. The existence of multiple caches (L1, L2, L3 and so on) in a system is an indicator of the potential performance of the system but there are some other essential variables that must also be considered. Specifications on the central processing units in our earlier consultative presentation must strictly be taken into account. Hence, the nature of the CPU must be fully appreciated in that it is where the cache process occurs. More importantly, the acquisition decision mut be based on the price, clock speed, number of cores, number of threads, size of the RAM, specification of graphics processing unit (GPU) and warranty offered by the vendor. Futuristically, the cache memory architecture in the supercomputer realm will likely spill over into the personal computer, mainframe and server domains as the technology develops and as users are driven by the pursuit of superior system performance.

References

1. Berger, Deanna, Jacobi, Christian, Walters, Craig R., Sonnelitter, Robert J., Cadigan, Mike, & Klein, Matthias. (2023). Enterprise-Class Multilevel Cache Design: Low Latency, Huge Capacity, and High Reliability. *IEEE MICRO*, *43*(1), 58–66. https://doi.org/10.1109/MM.2022.3193642
2. Brasler, Kevin. (2014). What you need to know before buying a new computer (Posted 2014-01-08 23:06:34). The Washington Post (Washington, D.C. 1974. Online).
3. Díaz Álvarez, Josefa, Colmenar, J. Manuel, Risco-Martín, José L., Lanchares, Juan, & Garnica, Oscar. (2015). Optimizing Performance of L1 Cache Memory for Embedded Systems driven by Differential Evolution. Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, 1383–1384. https://doi.org/10.1145/2739482.2764635.

4. Díaz Álvarez, Josefa, Colmenar, J. Manuel, Risco-Martín, José L., Lanchares, Juan, & Garnica, Oscar. (2015). Optimizing Performance of L1 Cache Memory for Embedded Systems driven by Differential Evolution. Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, 1383–1384. https://doi.org/10.1145/2739482.2764635
5. Fide, S., & Jenks, S. (2008). Proactive Use of Shared L3 Caches to Enhance Cache Communications in Multi-Core Processors. IEEE Computer Architecture Letters, 7(2), 57–60. https://doi.org/10.1109/L-CA.2008.10
6. Hameed, Fazal, & Castrillon, Jeronimo. (2019). A Novel Hybrid DRAM/STT-RAM Last-Level-Cache Architecture for Performance, Energy, and Endurance Enhancement. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, *27*(10), 2375–2386. https://doi.org/10.1109/TVLSI.2019.2918385
7. Knerl, L. (January 26, 2021). What is Cache Memory in My Computer? https://www.hp.com/us-en/shop/tech-takes/what-is-cache-memory.
8. Mano, M. (1993). Computer System Architecture. Prentice-Hall International (UK).
9. Parrilla-Gutierrez, Juan Manuel, Sharma, Abhishek, Tsuda, Soichiro, Cooper, Geoffrey J. T., Aragon-Camarasa, Gerardo, Donkers, Kevin, & Cronin, Leroy. (2020). A programmable chemical computer with memory and pattern recognition. *Nature Communications*, *11*(1), 1442–1442. https://doi.org/10.1038/s41467-020-15190-3
10. Rusu, S., Muljono, H., & Cherkauer, B. (2004). Itanium 2 processor 6M: higher frequency and larger L3 cache. IEEE MICRO, 24(2), 10–18. https://doi.org/10.1109/MM.2004.1289279
11. Sibai, Fadi N. (2008). On the performance benefits of sharing and privatizing second and third-level cache memories in homogeneous multi-core architectures. *Microprocessors and Microsystems*, *32*(7), 405–412. https://doi.org/10.1016/j.micpro.2008.06.002
12. Stallings, W (2013). Computer organizations and architecture designing for performance (9th Edition). Peason.
13. Takahashi, Hironao, Mori, Kinji, & Ahmad, Hafiz Farooq. (2010). Efficient I/O Intensive Multi-Tenant SaaS System Using L4 Level Cache. *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*, 222–228. https://doi.org/10.1109/SOSE.2010.14\
14. Wang, S.P. and Ledley, R.S. (2014) *Computer Architecture and security: Fundamentals of designing secure computer systems*. Hoboken: John Wiley & Sons.
15. Yadin, A. (2020). *Computer Systems Architecture*. Crc Press